What is claimed is:

1. A system for detecting an operating system exploitation which is of a type that renders a computer insecure, said system comprising:

(a) a storage device;

(b) an output device; and

(c) a processor programmed to:

(1) monitor the operating system to ascertain an occurrence of anomalous activity resulting from operating system behavior which deviates from any one of a set of pre-determined operating system parameters, wherein each of said pre-determined operating system parameters corresponds to a dynamic characteristic associated with an unexploited said operating system; and

(2) generate output on said output device which is indicative of any said anomalous activity that is ascertained.

2. A system according to claim 1 wherein the set of pre-determined operating system parameters is selected from:

(1) a first parameter corresponding to a requirement that all calls within the system call table associated with the operating system's kernel reference an address that is within the kernel's memory range;

(2) a second parameter corresponding to a requirement that each address range between adjacent modules in a linked list of modules be devoid of any active memory pages;

(3) a third parameter corresponding to a requirement that a kernel space view of each running process correspond to a user space view of each running process;

(4) a fourth parameter corresponding to a requirement that there be a capability to bind to any unused port on the computer; and

(5) a fifth parameter corresponding to a requirement that a kernel space view of each existing file correspond to a user space view of each existing file.

3. A system according to claim 2 wherein the operating system is Unix-based and the kernel memory range is between a starting address of 0xc0100000 and an ending address as determined with reference to one of a global variable and an offset calculation based on said global variable.

4. A system according to claim 1 wherein said processor is programmed to ascertain an occurrence of anomalous activity upon detecting any one of:

(a) a call within a system call table associated with the operating system's kernel which references a memory address outside of the kernel's memory range;

(b) an active memory page located within an address range between a pair of linked kernel modules, wherein said active memory page contains a module structure;

(c) a lack of correspondence between a kernel space view of each running process and a user space view of each running process;

(d) an inability to bind to an unused port on the computer; and

(e) a lack of correspondence between a kernel space view of each existing file and a user space view of each existing file.

5. A system according to claim 1 wherein said exploitation is selected from a group of comprises consisting of a hidden kernel module, a hidden system call table patch, a hidden process, a hidden file and a hidden port listener.

6. A system for detecting an operating system exploitation which is of a type that renders a computer insecure, said system comprising:

(a) storage means;

(b) output means;

(c) processing means for:

(1) monitoring the operating system to ascertain an occurrence of any anomalous activity resulting from behavior which deviates from any one of a set of pre-determined operating system parameters, wherein each of said pre-determined operating system parameters corresponds to a dynamic characteristic associated with an unexploited said operating system; and

(2) generating output on said output means which is indicative of any anomalous activity that is ascertained.

7. A computerized method for detecting exploitation of a computer operating system, comprising:

(a) establishing a set of operating system parameters, each corresponding to a dynamic characteristic associated with an unexploited operating system;

(b) monitoring the operating system to ascertain an occurrence of any anomalous activity resulting from behavior which deviates from any one of the set of operating system parameters; and

(c) generating output indicative of a detected exploitation upon ascertaining said anomalous activity.

8. A computerized method according to claim 7 wherein the set of operating system parameters is selected from a group consisting of:

(1) a first parameter corresponding to a requirement that all calls within the system call table associated with the operating system's kernel reference an address that is within the kernel's memory range;

(2) a second parameter corresponding to a requirement that each address range between adjacent modules in a linked list of modules be devoid of any active memory pages;

(3) a third parameter corresponding to a requirement that a kernel space view of each running process correspond to a user space view of each running process;

(4) a fourth parameter corresponding to a requirement that there be a capability to bind to any unused port on the computer; and

(5) a fifth parameter corresponding to a requirement that a kernel space view of each existing file correspond to a user space view of each existing file.

9. A computerized method according to claim 7 whereby a deviation is deemed to exist upon ascertaining any one of:

(a) a call within a system call table associated with the operating system's kernel which references a memory address outside of the kernel's memory range;

(b) an active memory page located within an address range between a pair of linked kernel modules, wherein said active memory page contains a module structure;

(c) a lack of correspondence between a kernel space view of each running process and a user space view of each running process;

(d) an inability to bind to an unused port on the computer; and

(e) a lack of correspondence between a kernel space view of each existing file and a user space view of each existing file.

10. A computerized method according to claim 7 wherein said exploitation is selected from a group of comprises consisting of a hidden kernel module, a hidden system call table patch, a hidden process, a hidden file and a hidden port listener.

11. A computerized method for detecting exploitation of a selected type of operating system, wherein the exploitation is one which renders a computer insecure, and whereby said method is capable of detecting said exploitation irrespective of whether the exploitation is signature-based and without a prior baseline view of the operating system, said method comprising:

monitoring the operating system to ascertain an occurrence of any anomalous activity resulting from behavior which deviates from any one of a set of operating system parameters, each operating system parameter corresponding to a dynamic characteristic associated with an unexploited operating system of the selected type.

12. A computerized method according to claim 11 whereby a deviation is deemed to exist upon ascertaining any one of:

(a) a call within a system call table associated with the operating system's kernel which references a memory address outside of the kernel's memory range;

(b) an active memory page located within an address range between a pair of linked kernel modules, wherein said active memory page contains a module structure;

(c) a lack of correspondence between a kernel space view of each running process and a user space view of each running process;

(d) an inability to bind to an unused port on the computer; and

(e) a lack of correspondence between a kernel space view of each existing file and a user space view of each existing file.

13. A computerized method according to claim 11 wherein said exploitation is selected from a group of comprises consisting of a hidden kernel module, a hidden system call table patch, a hidden process, a hidden file and a hidden port listener.

14. A computer-readable medium for use in detecting rootkit installations on a computer running an operating system, said computer-readable medium comprising a loadable kernel module having executable instructions for performing a method comprising:

monitoring the operating system to ascertain an occurrence of any anomalous activity resulting from behavior which deviates from any one of a set of

dynamic operating system parameters, each operating system parameter corresponding to a dynamic characteristic associated with an unexploited operating system of the selected type.

15. A computer-readable medium according to claim 14 wherein the set of operating system parameters is selected from a group consisting of:

(1) a first parameter corresponding to a requirement that all calls within the system call table associated with the operating system's kernel reference an address that is within the kernel's memory range;

(2) a second parameter corresponding to a requirement that each address range between adjacent modules in a linked list of modules be devoid of any active memory pages;

(3) a third parameter corresponding to a requirement that a kernel space view of each running process correspond to a user space view of each running process;

(4) a fourth parameter corresponding to a requirement that there be a capability to bind to any unused port on the computer; and

(5) a fifth parameter corresponding to a requirement that a kernel space view of each existing file correspond to a user space view of each existing file.

16. A computer-readable medium according to claim 15 wherein said executable instructions are operative to ascertain a deviation upon occurrence of any one of:

(a) a call within a system call table associated with the operating system's kernel which references a memory address outside of the kernel's memory range;

(b) an active memory page located within an address range between a pair of linked kernel modules, wherein said active memory page contains a module structure;

(c) a lack of correspondence between a kernel space view of each running process and a user space view of each running process;

(d) an inability to bind to an unused port on the computer; and

(e) a lack of correspondence between a kernel space view of each existing file and a user space view of each existing file.

17. A computer-readable medium for use in detecting a rootkit exploitation of a computer running a Linux operating system, wherein said rootkit exploitation is of a

type that renders the computer insecure, said computer-readable medium comprising:

(a) a loadable kernel module having executable instructions for performing a method comprising:

analyzing the operating system's memory to detect an existence of any hidden kernel module;

analyzing the operating system's system call table to detect an existence for any hidden patch thereto;

analyzing the computer to detect an existence of any hidden process; and

analyzing the computer to detect an existence of any hidden file.

18. A computer-readable medium according to claim 17 wherein said executable instructions are operative to analyze the computer for any hidden process by generating respective kernel space and user space views of running processes on the computer and ascertaining if a discrepancy exists therebetween.

19. A computer-readable medium according to claim 17 wherein said executable instructions are operative to analyze the computer for any hidden file by generating respective kernel space and user space views of existing files on the computer and ascertaining if a discrepancy exists therebetween.

20. A computer-readable medium according to claim 17 wherein said executable instructions are operative to analyze the system call table by initially obtaining an unbiased address for the system call table, and thereafter searching each call within the system call table to ascertain if it references an address outside of a dynamic memory range for the operating system's kernel.

21. A computer-readable medium according to claim 17 wherein said executable instructions are operative to display characteristic output results for any hidden kernel module, hidden system call table patch, hidden process and hidden file which is detected.